

# Лабораторная работа №6

## Построение и визуализация трёхмерных объектов

### Постановка задачи

Целью работы являлась разработка приложения для создания, визуализации и преобразования трёхмерных объектов. В соответствии с вариантом требовалось реализовать три этапа:

**Часть 6а:** Построение и визуализация трёхмерного объекта (каркасная модель буквы фамилии).

**Часть 6в:** Реализация трёхмерных преобразований объекта (масштабирование, перенос, вращение вокруг произвольной оси).

**Часть 6с:** Построение простейших проекций (три ортографические проекции на координатные плоскости).

### Теоретическая часть

#### Основы трёхмерной графики

**Трёхмерная графика** — область компьютерной графики, посвящённая методам создания изображений трёхмерных объектов. Основные концепции:

- **Вершины (vertices)** — точки в трёхмерном пространстве
- **Рёбра (edges)** — соединения между вершинами
- **Матричные преобразования** — операции над объектами в пространстве
- **Проекции** — преобразование 3D-объектов в 2D-изображение

#### Система координат

В работе использовалась декартова система координат:

- Ось X — горизонтальная ось (красный цвет)
- Ось Y — вертикальная ось (зелёный цвет)
- Ось Z — ось глубины (синий цвет)

## Матричные преобразования

Для выполнения преобразований использовались однородные координаты и матрицы  $4 \times 4$ :

- **Масштабирование:**

$$S = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- **Вращение вокруг осей:**

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_z(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- **Перенос:**

$$T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

## Порядок применения преобразований

Для корректного результата преобразования применяются в определённой последовательности:

$$P' = T \cdot R \cdot S \cdot P$$

где  $P$  — исходная точка,  $P'$  — преобразованная точка.

## Ортографические проекции

Ортографическая проекция — метод представления трёхмерного объекта на плоскости, при котором все проектирующие лучи параллельны:

- **Проекция на плоскость XY (вид сверху):**  $z = 0$
- **Проекция на плоскость XZ (вид спереди):**  $y = 0$
- **Проекция на плоскость YZ (вид сбоку):**  $x = 0$

# Реализация

## Архитектура приложения

Приложение разработано на C# с использованием Windows Forms. Основные компоненты:

- **Form1** — главная форма приложения
- **Panel** — холсты для отображения 3D-объекта и проекций
- **TrackBar** — элементы управления для преобразований
- **TextBox** — отображение матрицы преобразования
- **Button** — кнопки сброса параметров
- **Label** — подписи элементов управления

## Интерфейс приложения

### Основное окно

- **Основная область (слева)** — отображение 3D-объекта с осями координат
- **Область проекций (справа сверху)** — три ортографические проекции
- **Матрица преобразования (справа в центре)** — текущая матрица  $4 \times 4$
- **Элементы управления (снизу)** — трекбары для управления преобразованиями

### Элементы управления

Таблица 1: Элементы управления преобразованиями

Элемент	Параметр	Диапазон значений
Масштаб	Коэффициент масштабирования	10-200 (начальное: 50)
Вращение X	Угол вращения вокруг оси X	-180° до 180°
Вращение Y	Угол вращения вокруг оси Y	-180° до 180°
Вращение Z	Угол вращения вокруг оси Z	-180° до 180°
Перенос X	Смещение по оси X	-100 до 100
Перенос Y	Смещение по оси Y	-100 до 100
Перенос Z	Смещение по оси Z	-100 до 100

### Проекции

Каждая проекция имеет свои элементы управления:

- **Масштаб** — увеличение/уменьшение проекции
- **Смещение X** — горизонтальное перемещение
- **Смещение Y** — вертикальное перемещение

# Результаты работы

## Основные возможности

1. **Визуализация 3D-объекта:** Каркасная модель буквы К с цветовым кодированием (синие рёбра, красные вершины)
2. **Преобразования объекта:**
  - Масштабирование с сохранением пропорций
  - Вращение вокруг трёх осей координат
  - Перенос в любом направлении
3. **Ортографические проекции:**
  - Вид сверху (плоскость XY)
  - Вид спереди (плоскость XZ)
  - Вид сбоку (плоскость YZ)
4. **Отображение матрицы:** Текущая матрица преобразования в реальном времени

## Трудности реализации

### Технические сложности

1. **Преобразование систем координат:** Необходимость инвертирования оси Y для корректного отображения в экранных координатах
2. **Управление матрицами:** Корректный порядок умножения матриц для получения правильного результата
3. **Оптимизация отрисовки:** Эффективное обновление изображения при изменении параметров
4. **Интерфейс пользователя:** Организация множества элементов управления на одной форме

## Особенности реализации

### Для трёхмерных преобразований:

- Использование однородных координат для объединения преобразований
- Корректная обработка вращения в градусах и радианах
- Оптимизация вычислений матричных операций

### Для проекций:

- Независимое масштабирование и смещение для каждой проекции
- Корректное отображение осей координат на проекциях
- Синхронизация обновления всех областей отображения

# Тестирование

## Методы тестирования

1. **Визуальная проверка:** Контроль корректности отображения объекта при различных преобразованиях
2. **Проверка матриц:** Верификация правильности матричных операций через сравнение с эталонными значениями
3. **Границные случаи:** Тестирование при крайних значениях параметров
4. **Интеграционное тестирование:** Проверка взаимодействия всех компонентов приложения

## Результаты тестирования

Таблица 2: Результаты тестирования функциональности

Функция	Ожидаемый результат	Фактический результат
Масштабирование объекта	Пропорциональное изменение размера	Корректно работает
Вращение вокруг осей	Плавное вращение объекта	Корректно работает
Перенос объекта	Смещение объекта в пространстве	Корректно работает
Отображение проекций	Три ортографические проекции	Корректно отображаются
Обновление матрицы	Текущая матрица $4 \times 4$	Корректно обновляется
Элементы управления	Плавное изменение параметров	Корректно работают

## Выводы

В ходе выполнения лабораторной работы были достигнуты следующие результаты:

## Теоретические достижения

- Изучены основы трёхмерной компьютерной графики
- Освоены методы матричных преобразований в трёхмерном пространстве
- Поняты принципы ортографического проецирования
- Изучены алгоритмы визуализации каркасных моделей

## **Практические результаты**

- Разработано полнофункциональное приложение для визуализации 3D-объектов
- Реализована каркасная модель буквы К с 20 вершинами и 36 рёбрами
- Создана система трёхмерных преобразований (масштабирование, вращение, перенос)
- Реализованы три ортографические проекции с независимым управлением
- Создан интуитивно понятный пользовательский интерфейс
- Обеспечено отображение матрицы преобразования в реальном времени

## **Образовательная ценность**

- Приобретение навыков работы с трёхмерной графикой
- Опыт реализации матричных операций на C#
- Понимание принципов проецирования 3D-объектов на плоскость
- Навыки разработки сложных графических интерфейсов
- Опыт отладки геометрических алгоритмов

## **Перспективы развития**

1. **Добавление новых объектов:** Возможность загрузки различных 3D-моделей
2. **Улучшение визуализации:**
  - Добавление заливки граней
  - Реализация освещения и теней
  - Текстурирование объектов
3. **Расширение функциональности:**
  - Сохранение и загрузка состояний
  - Анимация преобразований
  - Экспорт в различные форматы
4. **Оптимизация:**
  - Использование аппаратного ускорения
  - Реализация Level of Detail (LOD)
  - Оптимизация матричных вычислений

# Заключение

Разработанное приложение успешно демонстрирует принципы трёхмерной графики и математических преобразований. Оно может использоваться как для учебных целей при изучении компьютерной графики, так и в качестве основы для более сложных систем трёхмерной визуализации.

Реализованные алгоритмы показали свою эффективность и соответствие теоретическим ожиданиям. Особенno важным достижением является корректная работа всех преобразований и возможность независимого управления проекциями.

Приложение успешно решает поставленные задачи лабораторной работы и предоставляет удобный инструмент для изучения основ трёхмерной компьютерной графики.

## Приложение: Примеры кода

```
1 private void DrawProjection(Graphics g, Panel panel,
2                             Func<Point3D, PointF> projection,
3                             string title)
4 {
5     g.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.AntiAlias;
6     g.Clear(Color.White);
7
8     using (var font = new Font("Arial", 10, FontStyle.Bold))
9     using (var brush = new SolidBrush(Color.Black))
10    {
11        g.DrawString(title, font, brush, 5, 5);
12    }
13
14    DrawProjectionAxes(g, panel.Width, panel.Height, title);
15
16    Point3D[] transformedVertices = new Point3D[vertices.Length];
17    for (int i = 0; i < vertices.Length; i++)
18    {
19        transformedVertices[i] = TransformPoint(vertices[i]);
20    }
21
22    using (Pen edgePen = new Pen(Color.DarkGreen, 1))
23    {
24        for (int i = 0; i < edges.GetLength(0); i++)
25        {
26            PointF p1 = projection(transformedVertices[edges[i, 0]]);
27            PointF p2 = projection(transformedVertices[edges[i, 1]]);
28            g.DrawLine(edgePen, p1, p2);
29        }
30    }
31 }
```

Листинг 1: Метод рисования проекций